CLAIMS

What is claimed is:

1. A reduced instruction set computer architecture implemented on a programmable logic device, comprising:

a parallel bit shifter capable of reversible shifts and bit reversals;

a Reed-Muller Boolean unit coupled to the parallel bit shifter; and

an immediate instruction function that, via constant modes, variously manipulates distribution of a set of literal bits of a half-word literal field from an instruction word across a full-length data word.

2. The reduced instruction set computer architecture of claim 1, wherein the parallel bit shifter is a parallel 32-bit shifter.

3. The reduced instruction set computer architecture of claim 1, wherein the parallel bit shifter comprises logic ranked into four sections.

4. The reduced instruction set computer architecture of claim 3, wherein the parallel bit shifter further comprises a split shift direction control signal applied on at least two of the four sections.

5. The reduced instruction set computer architecture of claim 1, wherein the Reed-Muller Boolean unit performs any Boolean operation with two operands and an inverse of the Boolean operation on data words by changing only a single control bit value.

6. The reduced instruction set computer architecture of claim 1, wherein the Reed-Muller Boolean unit multiplexes between two input operands by transposition of lone control bits.

7. The reduced instruction set computer architecture of claim 1, wherein the immediate instruction function treats the half-word literal field as one among a lower-half word, an upper-half word, a zero-filled word, a one-filled word, a sign-extended word, or a replicated 1-bit for 2-bits word.

8. The reduced instruction set computer architecture of claim 7, wherein the constant modes of the immediate instruction function include composite immediate instructions selected from the group of instructions comprising AND_FILL_LOW, OR_LOW, XOR_LOW, ADD_LOW, AND_FILL_HIGH, OR_HIGH, XOR_HIGH, ADD_HIGH, AND_DUPLEX, OR_DUPLEX, XOR_DUPLEX, ADD_DUPLEX, AND_SIGN, OR_SIGN, XOR_SIGN, and ADD_SIGN.

9. A system-on-chip, comprising:

a reduced instruction set computer processor implemented on a field programmable gate array fabric; and

a simple and balanced instruction set utilizing a minimal amount of resources from the field programmable gate array fabric, wherein the processor is synthesizable from hardware description language,

wherein the instruction set consists of 32 instructions, and wherein each instruction of the instruction set is a same size.

10. The system-on-chip of claim 9, wherein the same size is 32 bits.

11. The system-on-chip of claim 9, wherein one instruction of the instruction set is a reserved for future use (rfu) instruction.

12. The system-on-chip of claim 9, wherein a series of bitfield layouts of machine words are arranged and constructed for compatibility with on-chip intellectual property cores.

13. The system-on-chip of claim 12, wherein the series of bitfield layouts are arranged and constructed for use with standard processor buses.

14. The system-on-chip of claim 9, wherein the processor includes at least 16 general purpose registers and 7 special purpose registers.

15. The system-on-chip of claim 9, wherein the processor includes a sufficient number of registers for C compiler functionality yet minimal enough for efficient field programmable gate array (FPGA) realization.

16. The system-on-chip of claim 9, wherein bitfield layouts for instruction formats and instruction ordering are arranged for efficient use of FPGA resources.

17. The system-on-chip of claim 9, wherein instruction formats include a balance of single and paired instructions well-suited for compilation, such that after a first-issued instruction is considered by a compiler for issue, the compiler beneficially considers no more than 30 alternative instructions for a second-issued instruction to pair with the first-issued instruction.

18. The system-on-chip of claim 9, wherein instruction formats include a balance of single and paired instructions well-suited for a C runtime software environment, such that instruction sequences required to implement C language constructs naturally break into instruction sequences comprising one or two instructions.

19. The system-on-chip of claim 18, wherein the single and paired instructions minimize interrupt latency.

20. The system-on-chip of claim 18, wherein a minimal instruction overhead and the single and paired instructions

facilitate interrupt driver software design directly in the C language.

21. The system-on-chip of claim 9, wherein the processor is based on a 32-bit architecture.

22. The system-on-chip of claim 9, wherein the processor comprises at least one among a balanced instruction set; a set of six instruction forms; a byte-addressed 32-bit address space; addressing by word, half-word, or byte; a little endian byte ordering; and big endian byte ordering.

23. The system-on-chip of claim 8, wherein the processor is based on one among a 16-bit, a 32-bit, a 64-bit, a 128 bit, a 256-bit, a 512-bit and a 1024-bit architecture.

24. A method of forming a reduced instruction set computer processor, comprising the steps of:

embedding a processor core on a field programmable gate array; and

deploying a simple instruction set optimized for a compiler, wherein the processor is directly synthesizable from a hardware description language,

wherein the instruction set consists of 32 instructions, and wherein each instruction of the instruction set is a same size.

25. The method of claim 24, wherein one instruction of the instruction set is a reserved for future use (rfu) instruction.

26. The method of claim 24, wherein one instruction of the instruction set is an immediate instruction that, via constant modes, variously manipulates distribution of a set of literal bits of a half-word literal field from an instruction word across a full-length data word.

27. The method of claim 24, wherein embedding the processor core comprises configuring the field programmable gate array to include a parallel bit shifter capable of reversible shifts and bit reversals and a Reed-Muller Boolean unit coupled to the parallel bit shifter.

28. A reduced instruction set computer architecture implemented on a programmable logic device, comprising:

a parallel bit shifter capable of reversible shifts and bit reversals;

a Reed-Muller Boolean unit coupled to the parallel bit shifter; and

an immediate instruction function used in conjunction with the parallel bit shifter and Reed-Muller Boolean unit, wherein the immediate instruction function uses a single word instruction having N possible modes and having a plurality of instruction bits, with half of the plurality of instruction bits of the single word allocated to immediate data.

29. The reduced instruction set computer architecture of claim 28, wherein the immediate instruction function has 16 possible modes.

30. The reduced instruction set computer architecture of claim 28, wherein the immediate instruction function further includes a predetermined number of separate operation sub-modes.

31. The reduced instruction set computer architecture of claim 30, wherein the separate operation sub-modes are selected from the group of arithmetic logical operating modes comprising AND, OR, XOR, and ADD, and

wherein the AND mode is used for at least one of zeroing unwanted literal bits and masking of an operand, the OR mode is used for inserting desired literal bits into a selected general purpose register, the XOR mode is used for complementing select bits of an operand in a general purpose register when

necessary, and the ADD mode is used for immediate arithmetic operations.

32. The reduced instruction set computer architecture of claim 28, wherein the immediate instruction function further includes a predetermined number of separate bit mask sub-modes.

33. The reduced instruction set computer architecture of claim 32, wherein the separate bit mask sub-modes are selected from the group of bit mask sub-modes comprising FILL LOW, FILL HIGH, LOW, HIGH, DUPLEX, and SIGN bit masks.

34. The software reduced instruction set computer architecture of claim 33, wherein the LOW bit mask sub-mode is used for at least one of inserting literal bits into a lower half-word of a general purpose register and inserting as one of a pair of instructions for a full-word literal, the HIGH bit mask sub-mode is used for at least one of inserting literal bits into an upper half-word of a general purpose register and inserting as one of a pair of instructions for a full-word literal, the DUPLEX bit mask sub-mode is used for creating nybble, byte and half-word mask values flexibly, and the SIGN bit mask sub-mode is used for incrementing or decrementing a general purpose register by a constant value when combined with an ADD operating mode.

35. A system-on-chip, comprising:

a reduced instruction set computer processor implemented on a programmable logic device fabric;

a simple and balanced instruction set utilizing a minimal amount of resources from the field programmable gate array fabric, wherein the processor is synthesizable from hardware description language; and

a horizontally scalable immediate instruction using multiple vectorized versions of an N-bit architecture.

36. The system-on-chip of claim 35, wherein the horizontally scalable immediate instruction concurrently uses multiple vectorized versions of the N-bit architecture.

37. The system-on-chip of claim 35, wherein the N-bit architecture can be selected among multiple 16-bit, 32-bit, 64 bit, 128-bit, 256-bit, 512-bit and 1024-bit vectorized versions.

38. The system-on-chip of claim 35, wherein the N-bit architecture can be selected among multiple 18-bit, 36-bit, 72 bit, 144-bit, 288-bit, 576-bit and 1152-bit vectorized versions.

39. The system-on-chip of claim 38, wherein the reduced instruction set computer processor is a digital signal processor wherein the digital signal processor generates two N-bit words of constant data using an N/2 bit immediate instruction having N/4 bits dedicated for immediate data.

40. The system-on-chip of claim 39, wherein the two N-bit words comprise constants for a Fast Fourier Transform.

41. The system-on-chip of claim 39, wherein the two N-bit words comprise in-phase and quadrature data for complex signal processing.

42. The system-on-chip of claim 35, wherein the instruction set further comprises an reserved for future use (rfu) instruction that enables vertically scalable architectural variants implemented on the field programmable gate array fabric.

43. The system-on-chip of claim 42, wherein the rfu instruction enables upward compatible single and double-precision floating-point arithmetic operations via the field programmable gate array fabric.

44. The system-on-chip of claim 43, wherein the rfu instruction uses a single standard extension at a hardware/software boundary for variant architectures while minimally impacting a compiler design for the variant architectures.